ОНЛАЙН-ОБРАЗОВАНИЕ

Меня хорошо видно && слышно?

Ставьте + , если все хорошо
Напишите в чат, если есть проблемы

Flutter Mobile Developer

Тема 9. Dart. Isolates

Цель урока

- научиться работать со Isolates

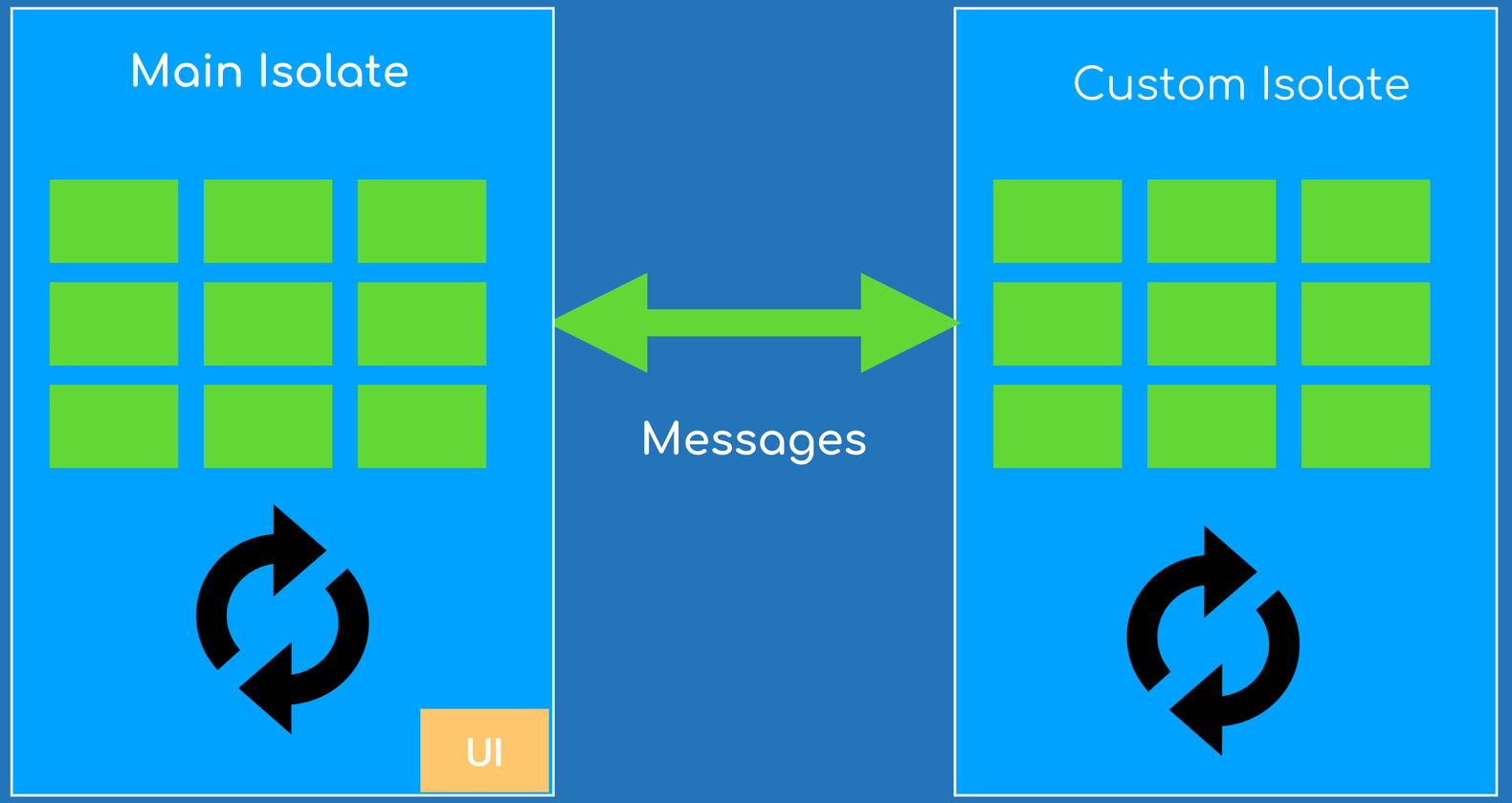
О чем будем говорить

Isolates - Lifecycle

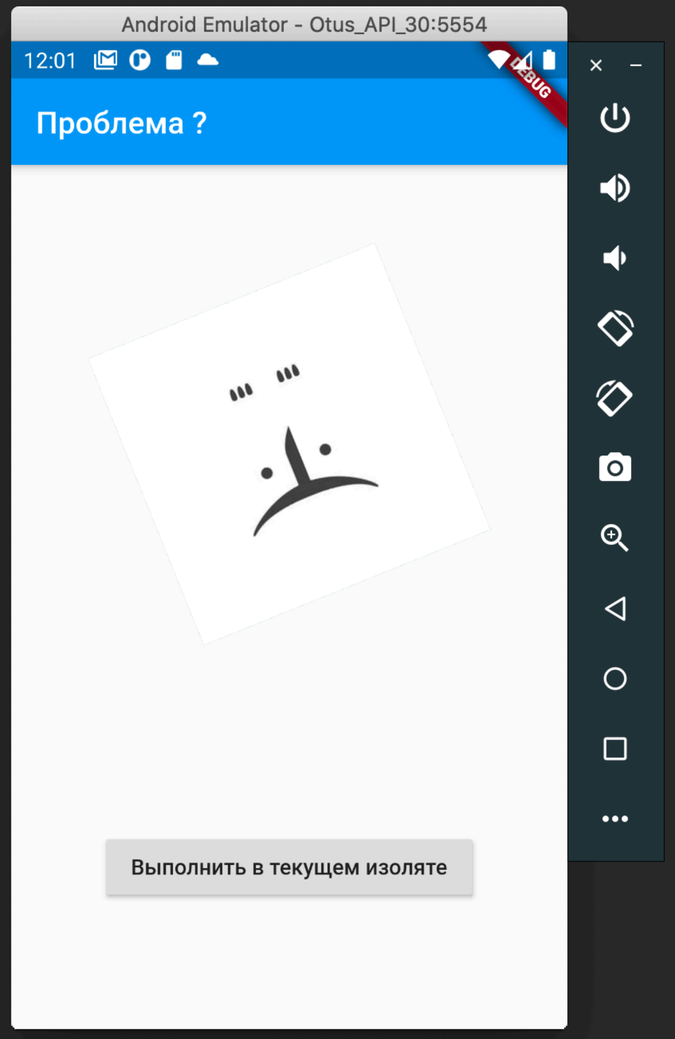
Isolates - Обмен данными

Flutter Application

В общем случае приложение может содержать несколько изолятов



Problem


Android Emulator - Otus_API_30:5554
12:01
DEBUG
Проблема ?
Выполнить в текущем изоляте

Что такое Isolate ?

Dart › dart:isolate library

https://api.dart.dev/stable/2.10.3/dart-isolate/dart-isolate-library.html

Dart SDK

LIBRARIES

CORE

- dart:async
- dart:collection
- dart:convert
- dart:core
- dart:developer
- dart:math
- dart:typed_data

VM

- dart:ffi
- dart:io
- dart:isolate

dart:isolate library

Null safety

Concurrent programming using *isolates*: independent workers that are similar to threads but don't share memory, communicating only via messages.

To use this library in your code:

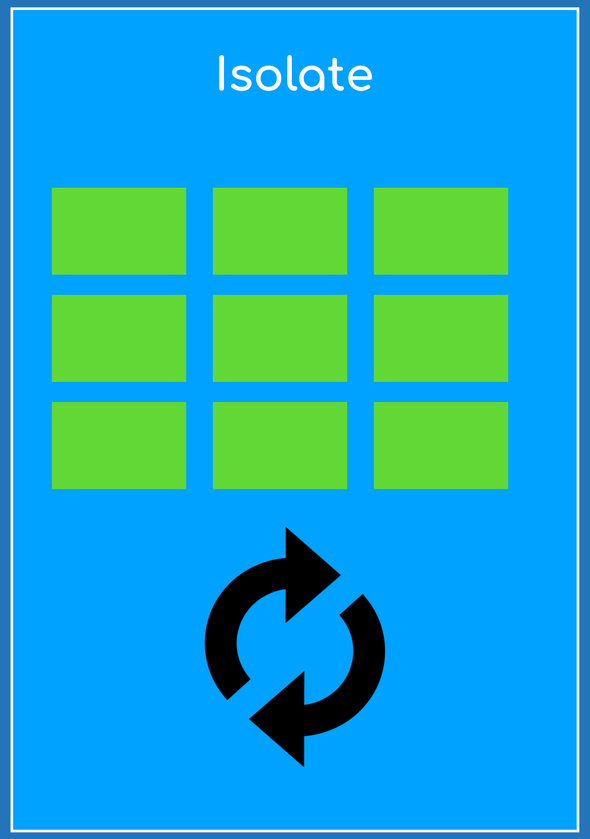
```
import 'dart:isolate';
```

Classes

Capability
An unforgeable object that comes back as equal when passed through other isolates. [...]

Isolate
An isolated Dart execution context. [...]

Что такое Isolate ?



- Способ запустить другой код параллельно, задействуя все мощности процессора
- Нет общей памяти с другими изолятами
- Отдельный Event Loop
- Сообщения между изолятами копируются

Isolates - Lifecycle - Создаем Isolate

```
external static Future<Isolate> spawn<T>(void entryPoint(T message), T message,
    {… @Since("2.3") String? debugName});
```

Isolates - Lifecycle - Удаляем Isolate

```
external void kill({int priority = beforeNextEvent});


        /** Argument to `ping` and `kill`: Ask for immediate action. */
        static const int immediate = 0;


        /** Argument to `ping` and `kill`: Ask for action before the next event. */
        static const int beforeNextEvent = 1;
```

Isolates - Lifecycle - Demo

00_start.dart

```dart
  stdout.writeln('Создадим новый Isolate и передадим аргумент в entry point');
  Isolate isolate = await Isolate.spawn(someSoHeavyAndLongTask, 0);

  stdout.writeln('Здесь начинается выполнение главного изолята');
  new Timer.periodic(Duration(seconds: 1), (timer) {
    stdout.writeln('Значение таймер в главном изоляте : ${timer.tick}');
  }); // Timer.periodic

  ///
  /// Ожидаем нажатия в консоли и после этого убиваем наш новый изолят
  ///
```

.dart

Variables | Console

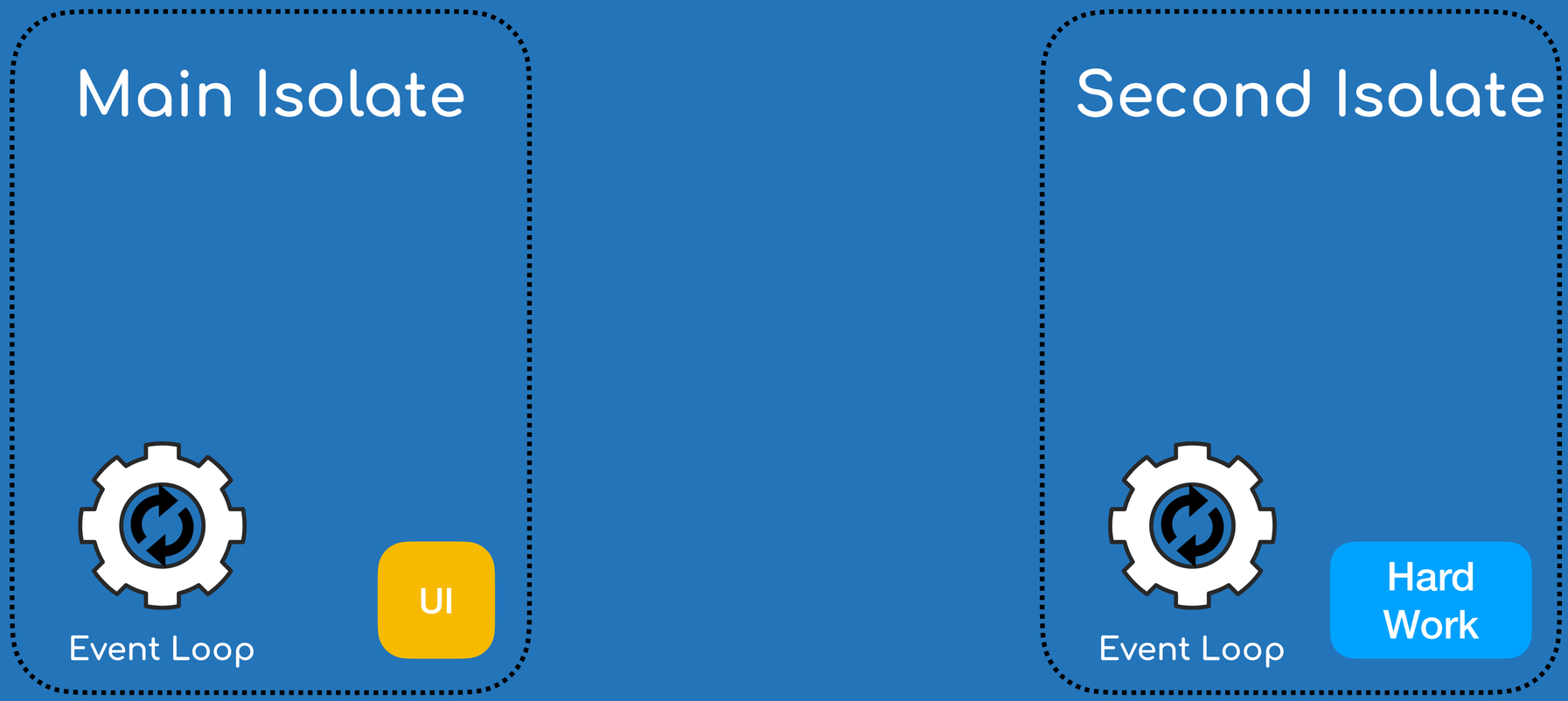
```
Значение таймер в новом изоляте : 32
Значение таймер в главном изоляте : 33
Значение таймер в новом изоляте : 33
Значение таймер в главном изоляте : 34
Значение таймер в новом изоляте : 34
Значение таймер в главном изоляте : 35
Значение таймер в новом изоляте : 35
```

О чем будем говорить

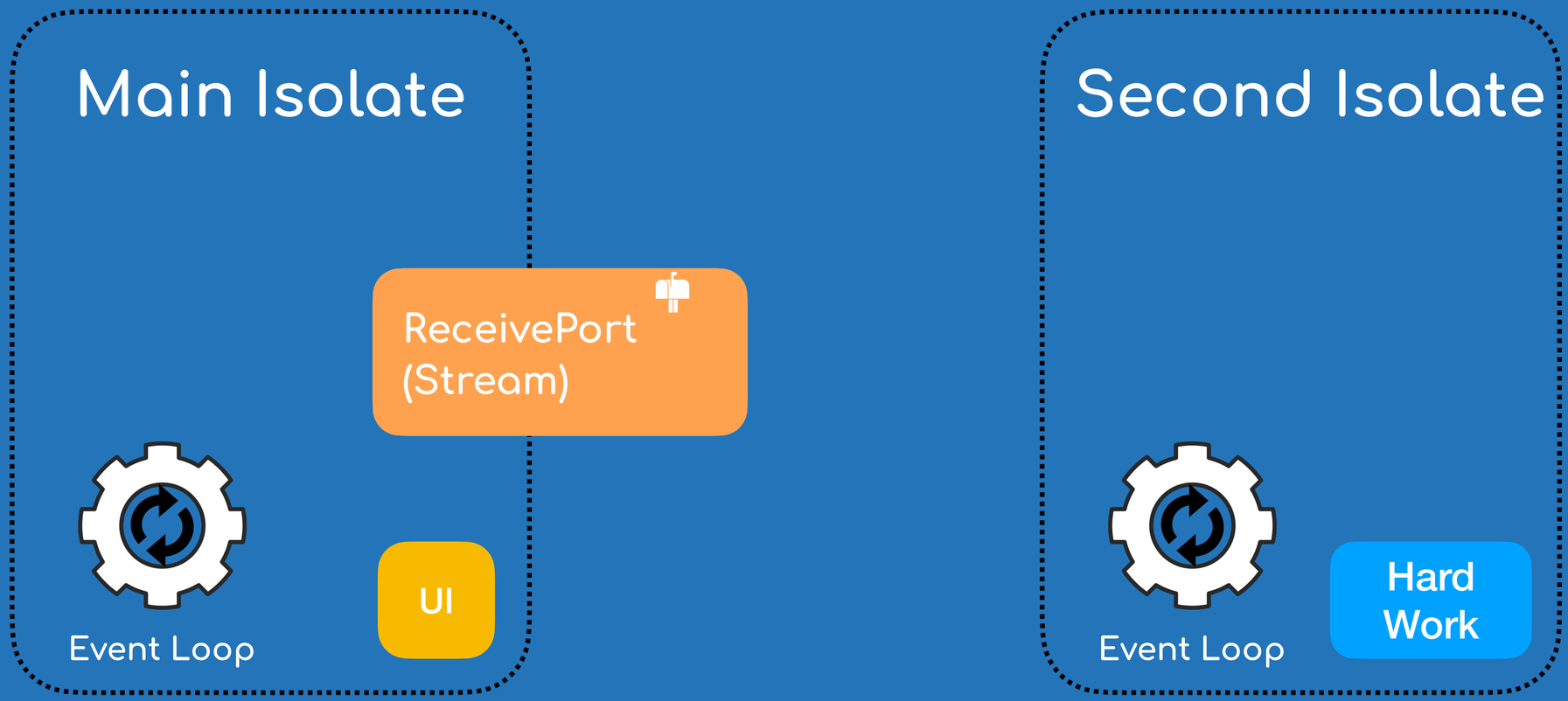
Isolates - Lifecycle

Isolates - Обмен данными

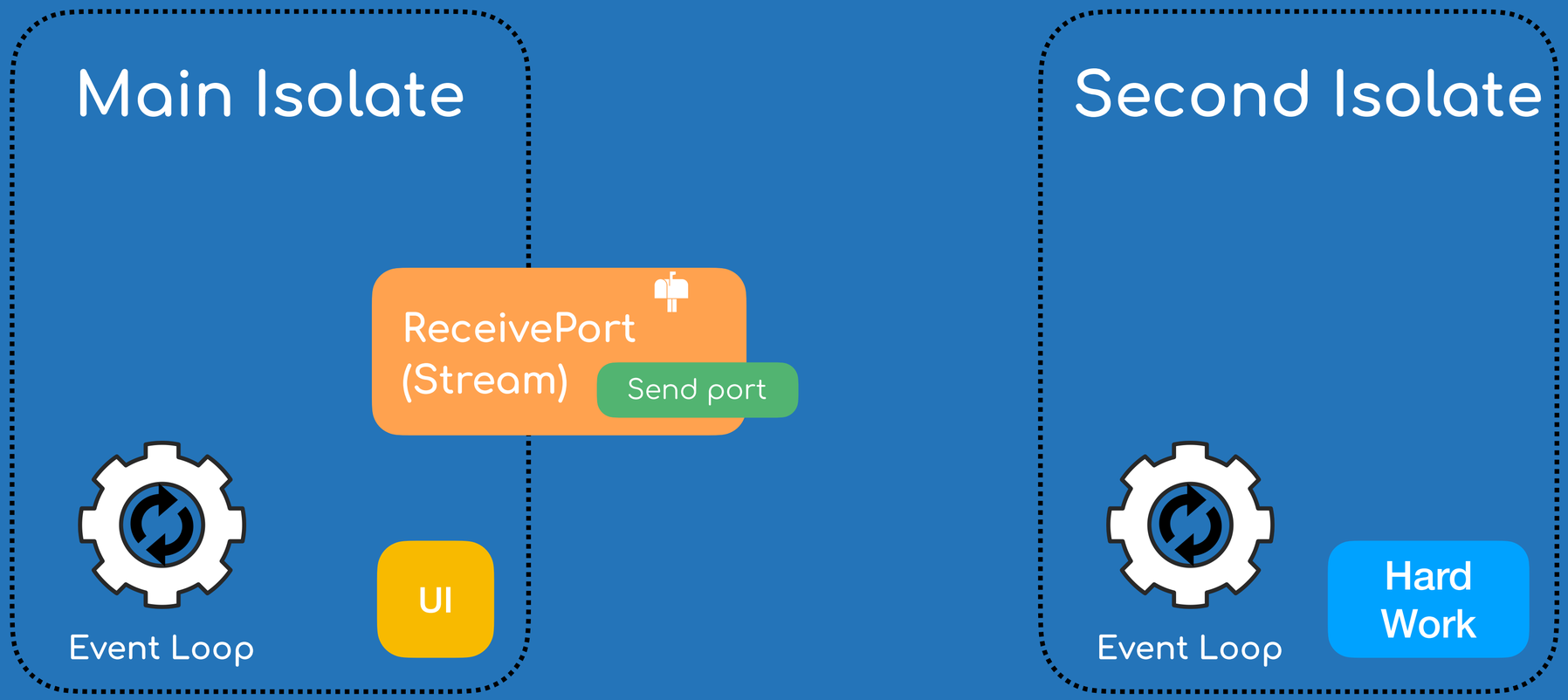
Взаимодействие между Isolates



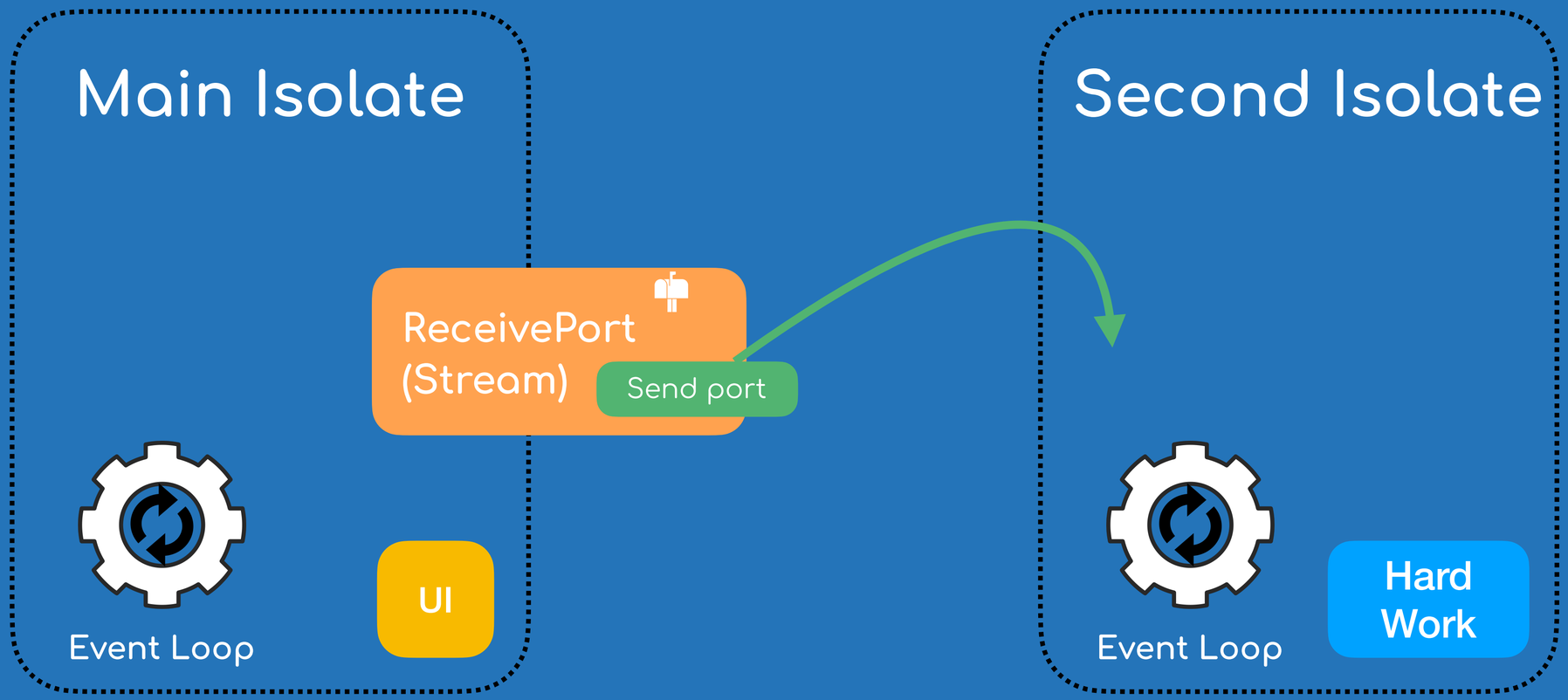
Взаимодействие между Isolates



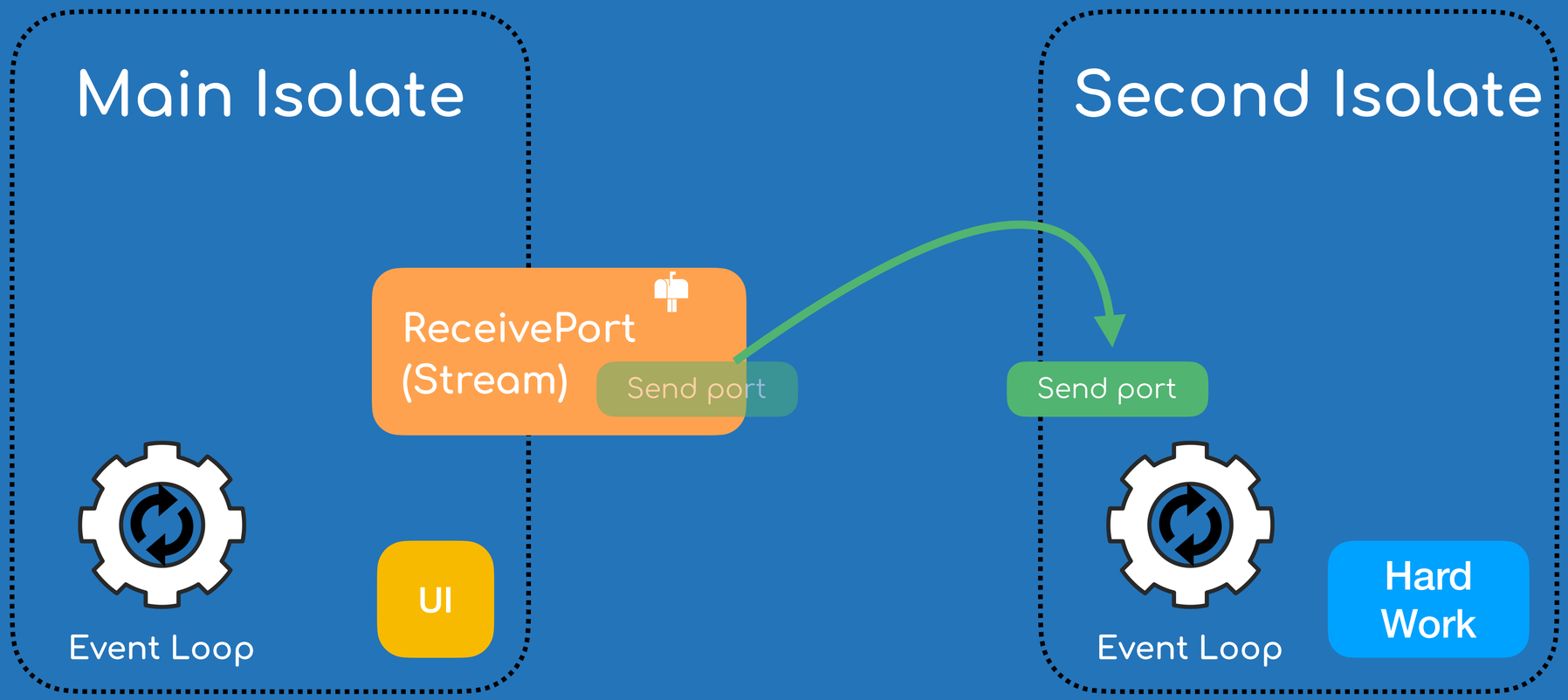
Взаимодействие между Isolates



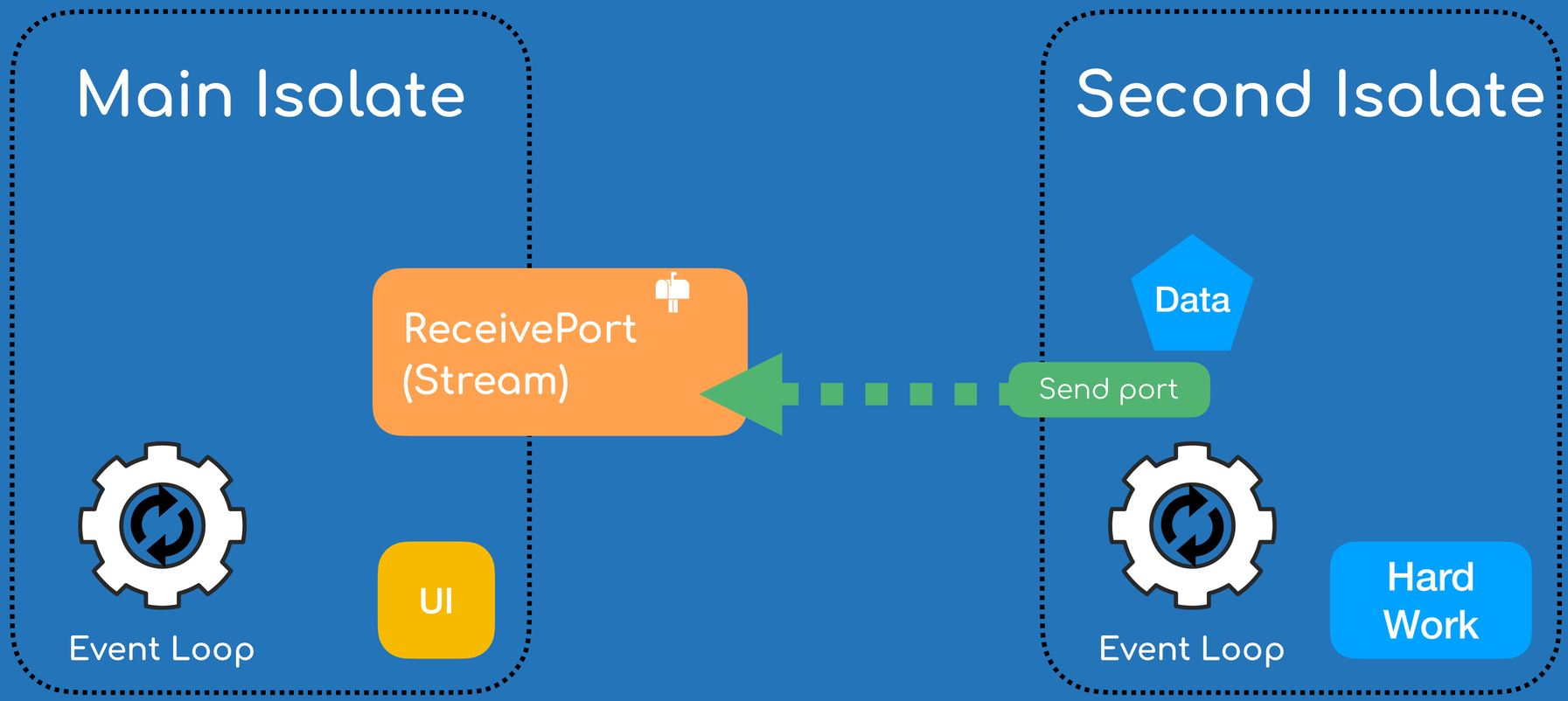
Взаимодействие между Isolates



Взаимодействие между Isolates



Взаимодействие между Isolates



Ports

```
/**
 * Together with [SendPort], the only means of communication between isolates.
 *
 * [ReceivePort]s have a `sendPort` getter which returns a [SendPort].
 * Any message that is sent through this [SendPort]
 * is delivered to the [ReceivePort] it has been created from. There, the
 * message is dispatched to the `ReceivePort`'s listener.
 *
 * A [ReceivePort] is a non-broadcast stream. This means that it buffers
 * incoming messages until a listener is registered. Only one listener can
 * receive messages. See [Stream.asBroadcastStream] for transforming the port
 * to a broadcast stream.
 *
 * A [ReceivePort] may have many [SendPort]s.
 */
abstract class ReceivePort implements Stream<dynamic> {
```

Ports

```dart
/**
 * Sends messages to its [ReceivePort]s.
 *
 * [SendPort]s are created from [ReceivePort]s. Any message sent through
 * a [SendPort] is delivered to its corresponding [ReceivePort]. There might be
 * many [SendPort]s for the same [ReceivePort].
 *
 * [SendPort]s can be transmitted to other isolates, and they preserve equality
 * when sent.
 */
abstract class SendPort implements Capability {
```

Ports

```
  * Sends an asynchronous [message] through this send port, to its
  * corresponding `ReceivePort`.
  *
  * The content of [message] can be: primitive values (null, num, bool, double,
  * String), instances of [SendPort], and lists and maps whose elements are any
  * of these. List and maps are also allowed to be cyclic.
  *
  * In the special circumstances when two isolates share the same code and are
  * running in the same process (e.g. isolates created via [Isolate.spawn]), it
  * is also possible to send object instances (which would be copied in the
  * process). This is currently only supported by the
  * [Dart Native](https://dart.dev/platforms#dart-native-vm-jit-and-aot)
  * platform.
  *
  * The send happens immediately and doesn't block.  The corresponding receive
  * port can receive the message as soon as its isolate's event loop is ready
  * to deliver it, independently of what the sending isolate is doing.
```

Взаимодействие между Isolates - Demo

01_one_way_pass_data_using_ports.dart

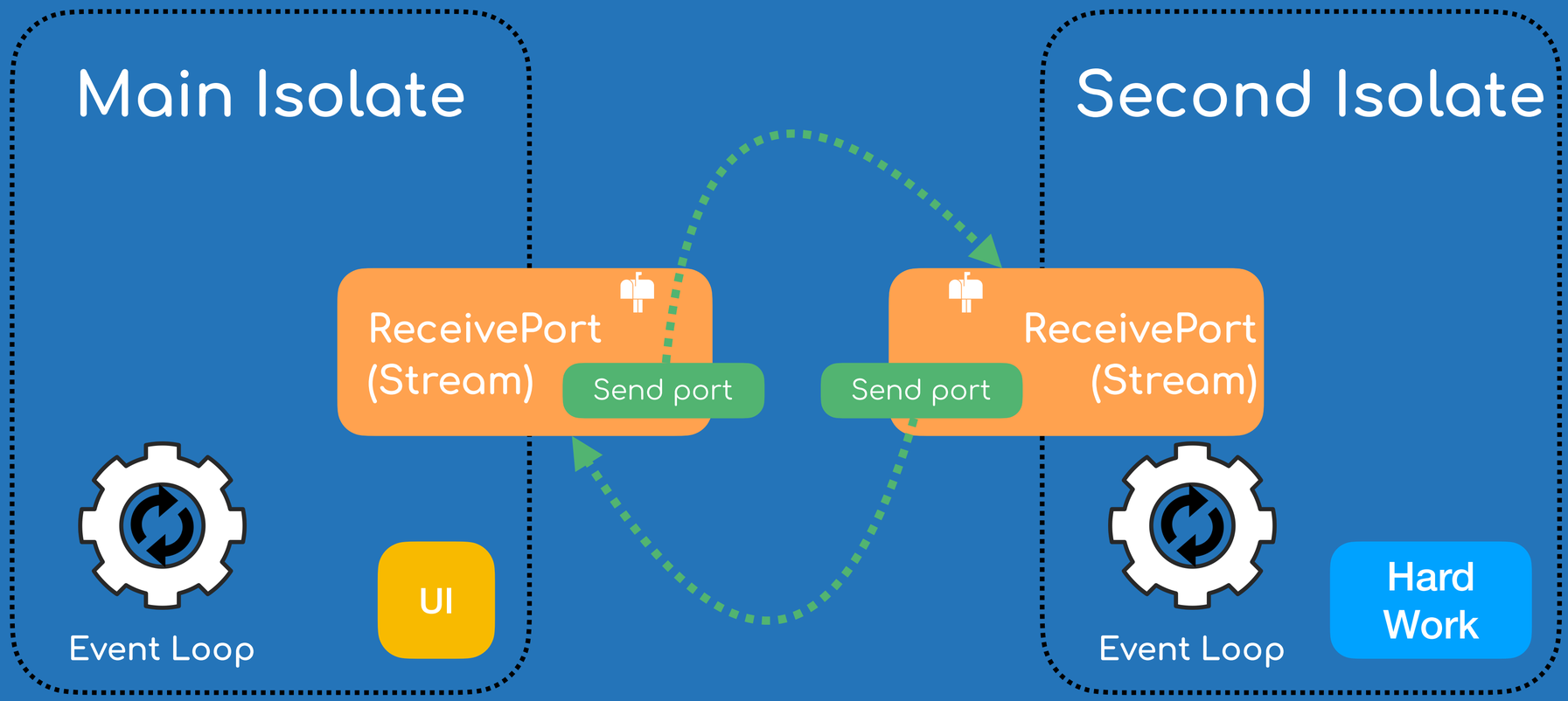
```
22    receivePort.listen((dynamic receivedData) {
23      stdout.writeln('Получены новые данные из нового изолята : $receivedData');
24
25      ///
26      /// Если мы получили сообщение
27      ///
28      if (receivedData is String && receivedData.toString().contains(taskCompletedMessage)) {
29        stdout.writeln('Получены сообщение $taskCompletedMessage, уничтожим новый изолят');
30        isolate.kill(priority: Isolate.immediate);
31        isolate = null;
32        receivePort.close();
```

ass_data_using_ports.dart

Variables | Console

```
Создадим новый Isolate и передадим аргумент в entry point
Здесь начинается выполнение главного изолята
Новый изолят создан с аргументом : SendPort
Значение progress в новом изоляте : 0
Значение progress в главном изоляте : 0
```

Двунаправленное взаимодействие



Взаимодействие между Isolates - Demo

02_isolates_listening_each_other_using_ports.dart

```dart
  stdout.writeln('Здесь начинается выполнение нового изолята [${Isolate.current.debugName}]');
  stdout.writeln('Новый изолят создан с аргументом : $sendPort');

  ///
  /// Внутри нового изолята мы создаем свой порт для приема сообщений.
  /// Но для того, чтобы начать принимать от другой стороны сообщения,
  /// мы должны расшарить send port
  ///
  ReceivePort receivePort = ReceivePort();
  receivePort.listen((dynamic receivedData) {
    stdout.writeln('Получены новые данные из главного изолята : $receivedData');
    stdout.writeln('task $taskCompletedMessage');

    sendPort.send(taskCompletedMessage);
  });
```

tening_each_other_using_ports...

Variables | Console

```
Получены новые данные из нового изолята : 7
Получены новые данные из нового изолята : 8
```

Другие альтернативы

StreamChannel

stream_channel

6 LIKES | 80 PUB POINTS | 98% POPULARITY

An abstraction for two-way communication channels.

v 2.0.0 / 2.1.0-nullsafety.2 • Published: Mar 28, 2019

DART | NATIVE | JS | FLUTTER | ANDROID | IOS | WEB

API results: ▶ stream_channel/stream_channel-library.html

streams_channel

9 LIKES | 100 PUB POINTS | 93% POPULARITY

StreamsChannel is inspired from EventChannel. It allows to create streams of events between Flutter and platform side.

v 0.3.0 • Published: Feb 5, 2020 loup.app

FLUTTER | ANDROID | IOS

API result: streams_channel/StreamsChannel-class.html

StreamChannel

stream_channel 2.0.0

Published Mar 28, 2019 • Latest: 2.0.0 / Prerelease: 2.1.0-nullsafety.2

DART | NATIVE | JS | FLUTTER | ANDROID | IOS | WEB

6

Readme | Changelog | Installing | Versions | Scores

This package exposes the `StreamChannel` interface, which represents a two-way communication channel. Each `StreamChannel` exposes a `Stream` for receiving data and a `StreamSink` for sending it.

`StreamChannel` helps abstract communication logic away from the underlying protocol. For example, the `test` package re-uses its test suite communication protocol for both WebSocket connections to browser suites and Isolate connections to VM tests.

This package also contains utilities for dealing with `StreamChannel`s and with two-way communications in general. For documentation of these utilities, see the API docs.

6 LIKES | 80 PUB POINTS | 98% POPULARITY

Metadata

An abstraction for two-way communication channels.

Repository (GitHub)

View/report issues

Documentation

API reference

Uploaders

StreamChannel - Demo

04_isolates_listening_each_other_using_channel.dart

```
20    channel.stream.listen((receivedData) {
21      stdout.writeln('Получены новые данные из нового изолята : $receivedData');
22
23      if (receivedData == 'hi') {
24        channel.sink.add('how are you?');
25      } else if ((receivedData as String).contains('bye')) {
26        channel.sink.add('have to kill you...');
27        stdout.writeln('have to kill you...');
28
29        channel.sink.close();
30        isolate.kill(priority: Isolate.immediate);
31      }
32    });
33  }
34
35  ///
```

ng_each_other_using_chan...

Variables Console

Custom implementation

05_custom_sugar.dart

```dart
import 'dart:async';
import 'dart:io';
import 'dart:isolate';

void main() async {
  stdout.writeln('Текущий isolate: ${Isolate.current.debugName}');
  SendPort mainToIsolateStream = await compute(
    isolatedMethod: myIsolate,
    messageCallBack: (data) => stdout.writeln('Коллбек : $data'),
    debugName: 'Debug isolate',
  );

  mainToIsolateStream.send('Сообщение 1 из \'${Isolate.current.debugName}\' изолята');
  mainToIsolateStream.send('Сообщение 2 из \'${Isolate.current.debugName}\' изолята');
  mainToIsolateStream.send('kill');
```

ugar.dart

Variables | Console

```
Коллбек : Это сообщение 1 из 'Debug isolate' изолята
Коллбек : Это сообщение 2 из 'Debug isolate' изолята
Коллбек : Это сообщение 3 из 'Debug isolate' изолята
```

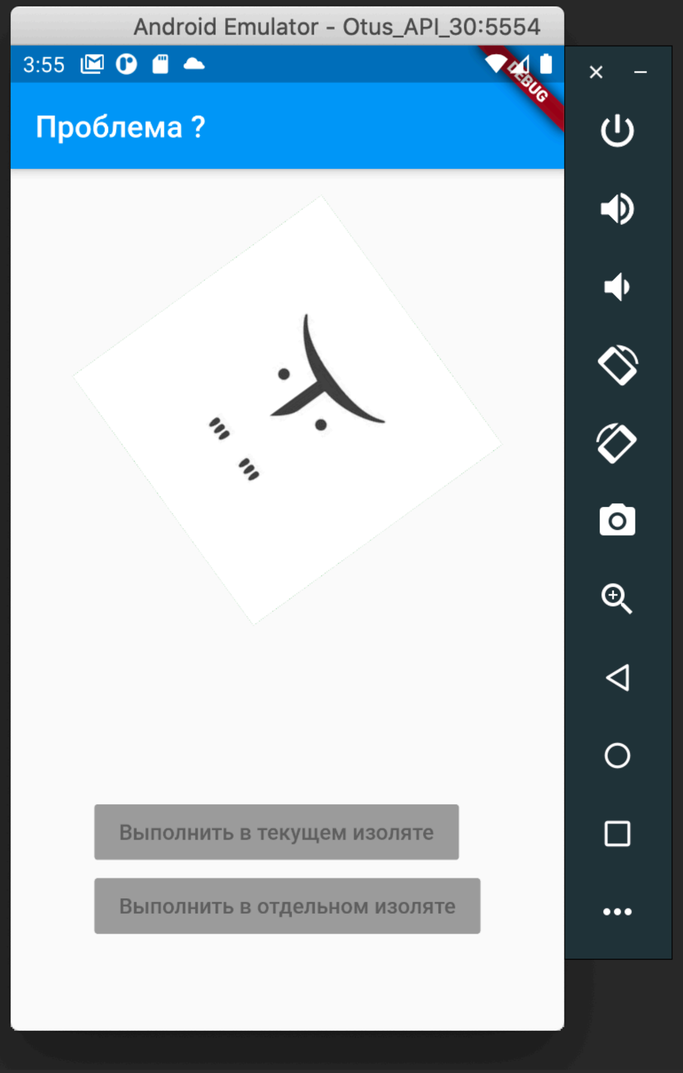
Flutter implementation

```dart
/// The dart:io implementation of [isolate.compute].
Future<R> compute<Q, R>(isolates.ComputeCallback<Q, R> callback, Q message,
{ String? debugLabel }) async {
```

compute()

```dart
/// Spawn an isolate, run `callback` on that isolate, passing it `message`, and
/// (eventually) return the value returned by `callback`.
///
/// This is useful for operations that take longer than a few milliseconds, and
/// which would therefore risk skipping frames. For tasks that will only take a
/// few milliseconds, consider [SchedulerBinding.scheduleTask] instead.
///
/// {@template flutter.foundation.compute.types}
/// `Q` is the type of the message that kicks off the computation.
///
/// `R` is the type of the value returned.
/// {@endtemplate}
///
/// The `callback` argument must be a top-level function, not a closure or an
/// instance or static method of a class.
///
/// {@template flutter.foundation.compute.limitations}
/// There are limitations on the values that can be sent and received to and
/// from isolates. These limitations constrain the values of `Q` and `R` that
/// are possible. See the discussion at [SendPort.send].
/// {@endtemplate}
///
/// The `debugLabel` argument can be specified to provide a name to add to the
/// [Timeline]. This is useful when profiling an application.
// Remove when https://github.com/dart-lang/sdk/issues/37149 is fixed.
// ignore: prefer_const_declarations
final _ComputeImpl compute = _isolates.compute;
```

Решение проблемы - Demo



Передача сообщений - Demo

07_data_transfer.dart

```dart
13  void main() async {
14    ReceivePort receivePort = ReceivePort();
15
16    stdout.writeln('Создадим новый Isolate и передадим аргумент в entry point');
17    await Isolate.spawn(someSoHeavyAndLongTask, receivePort.sendPort);
18
19    final sourceData = Data('test message');
20    stdout.writeln('data hash is ${sourceData.hashCode}');
21
```

sfer.dart

Variables | Console

```
Создадим новый Isolate и передадим аргумент в entry point
data hash is 660487516
data from Main Isolate: Instance of 'Data'
data hash is 506029844
data from new isolate : Instance of 'Data'
data hash is 84979178
sourceData and receivedData is not equal
```

Передача сообщений - spawnByUri - Demo

08_data_transfer_spawn_by_uri.dart

```
6   void main() async {
7     ReceivePort receivePort = ReceivePort();
8
9     await Isolate.spawnUri(Uri.parse('./08_isolate_worker.dart'), [], receivePort.sendPort);
10
11    ///
12    /// comment this
13    ///
14    final sourceData = Data('test message');
```

Run: 08_data_transfer_spawn_by_uri.dart

```
dart 08_data_transfer_spawn_by_uri.dart [Debug service available at http://127.0.0.1:58835/wpTiJTnidDc=/]

lib/08_data_transfer_spawn_by_uri.dart: Warning: Interpreting this as package URI,
  'package:lesson_09/08_data_transfer_spawn_by_uri.dart'.
data hash is 246293711
Unhandled exception:
Invalid argument(s): Illegal argument in isolate message : (object is a regular Dart Instance)
#0      _SendPortImpl._sendInternal (dart:isolate-patch/isolate_patch.dart:219:70)
#1      _SendPortImpl.send (dart:isolate-patch/isolate_patch.dart:203:5)
#2      main.<anonymous closure> (package:lesson_09/08_data_transfer_spawn_by_uri.dart:26:28)
#3      _RootZone.runUnaryGuarded (dart:async/zone.dart:1384:10)
#4      _BufferingStreamSubscription._sendData (dart:async/stream_impl.dart:357:11)
#5      _BufferingStreamSubscription._add (dart:async/stream_impl.dart:285:7)
#6      _SyncStreamControllerDispatch._sendData (dart:async/stream_controller.dart:808:19)
#7      _StreamController._add (dart:async/stream_controller.dart:682:7) [2 more...]
```

TransferableTypedData

Since Dart 2.4 (currently in Flutter stable), you can send large amounts of data between isolates (i.e. CPU threads) cheaply by using TransferableTypedData.

TransferableTypedData

dart:isolate library

CLASSES
Capability
Isolate
RawReceivePort
ReceivePort
SendPort
TransferableTypedData

EXCEPTIONS
IsolateSpawnException
RemoteError

TransferableTypedData class Null safety

An efficiently transferable sequence of byte values.

A TransferableTypedData is created from a number of bytes. This will take time proportional to the number of bytes.

The TransferableTypedData can be moved between isolates, so sending it through a send port will only take constant time.

When sent this way, the local transferable can no longer be materialized, and the received object is now the only way to materialize the data.

Annotations
@Since("2.3.2")

Constructors

TransferableTypedData.fromList(List<TypedData> list)
Creates a new TransferableTypedData containing the bytes of `list`. [...]
factory

TransferableTypedData

```dart
/**
 * An efficiently transferable sequence of byte values.
 *
 * A [TransferableTypedData] is created from a number of bytes.
 * This will take time proportional to the number of bytes.
 *
 * The [TransferableTypedData] can be moved between isolates, so
 * sending it through a send port will only take constant time.
 *
 * When sent this way, the local transferable can no longer be materialized,
 * and the received object is now the only way to materialize the data.
 */
@Since("2.3.2")
abstract class TransferableTypedData {
  /**
   * Creates a new [TransferableTypedData] containing the bytes of [list].
   *
   * It must be possible to create a single [Uint8List] containing the
   * bytes, so if there are more bytes than what the platform allows in
   * a single [Uint8List], then creation fails.
   */
  external factory TransferableTypedData.fromList(List<TypedData> list);

  /**
   * Creates a new [ByteBuffer] containing the bytes stored in this [TransferableTypedData].
   *
   * The [TransferableTypedData] is a cross-isolate single-use resource.
   * This method must not be called more than once on the same underlying
   * transferable bytes, even if the calls occur in different isolates.
   */
  ByteBuffer materialize();
}
```

TransferableTypedData - Demo - Test

```
❯ time dart 09_data_transfer.dart -typeddata
09_data_transfer.dart: Warning: Interpreting this as package URI, 'package:lesson_09/09_data_transfer.dart'.
Создадим новый Isolate и передадим аргумент в entry point
current test: withTransferableTypedData is true
100000000
elapsed 26336
dart 09_data_transfer.dart -typeddata  3.55s user 0.89s system 110% cpu 4.001 total
❯ time dart 09_data_transfer.dart
09_data_transfer.dart: Warning: Interpreting this as package URI, 'package:lesson_09/09_data_transfer.dart'.
Создадим новый Isolate и передадим аргумент в entry point
current test: withTransferableTypedData is false
100000000
elapsed 4665758
dart 09_data_transfer.dart  6.74s user 1.08s system 98% cpu 7.912 total
```

Другие альтернативые реализации

isolate 2.0.3

Published Feb 4, 2020

DART | NATIVE | FLUTTER | ANDROID | IOS

18

Readme | Changelog | Installing | Versions | Scores

build passing

Helps with isolates and isolate communication in Dart. Requires the `dart:isolate` library being available. Isolates are not available for Dart on the web.

The package contains individual libraries with different purposes.

Creating send ports and responding to messages.

The "ports.dart" sub-library contains functionality for creating `SendPort`s and reacting to values sent to those ports.

Working with isolates and running functions in other isolates.

The "isolate_runner.dart" sub-library introduces an `IsolateRunner` class that gives easy access to the `Isolate` functionality, and also gives a way to run new functions in the isolate repeatedly, instead of just on the initial `spawn` call.

18 LIKES | 90 PUB POINTS | 95% POPULARITY

Metadata

Utility functions and classes related to the 'dart:isolate' library.

Repository (GitHub)

View/report issues

Documentation

API reference

Uploaders

lrn@google.com

kevmoo@google.com

Другие альтернативые реализации

Creating send ports and responding to messages.

The "ports.dart" sub-library contains functionality for creating `SendPort`s and reacting to values sent to those ports.

Working with isolates and running functions in other isolates.

The "isolate_runner.dart" sub-library introduces an `IsolateRunner` class that gives easy access to the `Isolate` functionality, and also gives a way to run new functions in the isolate repeatedly, instead of just on the initial `spawn` call.

A central registry for values that can be used across isolates.

The "registry.dart" sub-library provides a way to create an object registry, and give access to it across different isolates.

Balancing load across several isolates.

The "load_balancer.dart" sub-library can manage multiple `Runner` objects, including `IsolateRunner`, and run functions on the currently least loaded runner.

Features and bugs

Please file feature requests and bugs at the issue tracker.

View/report issues

Documentation

API reference

Uploaders

lrn@google.com

kevmoo@google.com

License

BSD (LICENSE)

More

Packages that depend on isolate

IsolateRunner - Demo

10_isolate_runner.dart

```
7   stdout.writeln('Текущий изолят ${Isolate.current.debugName}');
8
9   final Runner runner = await IsolateRunner.spawn();
10  final output = await runner.run<String, int>(someSoHeavyAndLongTask, 5);
11
12  //...
17
18  stdout.writeln(output);
```

Run: 10_isolate_runner.dart

```
Текущий изолят main
Текущий изолят _create
Новый изолят создан с аргументом : 5
Значение progress в новом изоляте : 0
Значение progress в новом изоляте : 1
Значение progress в новом изоляте : 2
Значение progress в новом изоляте : 3
Значение progress в новом изоляте : 4
It's all ok!

Process finished with exit code 0
```

Выводы - Isolates - это хорошо?

- Способ запустить другой код параллельно, задействуя все мощности процессора
- Нет общей памяти с другими изолятами (это не означает, что вы можете открыть один файл в двух изолятах)
- Отдельный Event Loop
- Сообщения между изолятами копируются

Все ли хорошо с Isolates?

- Используем Isolate - это плюс ~ 1.5-2mb зависимостей в коде
- Запуск Isolate не бесплатный - запуск занимает порядка ~100ms
- При обмене сообщениями происходит постоянное копирование данных
- Нельзя использовать platform channels (альтернатива - использовать FFI или https://pub.dev/packages/flutter_isolate (не проверял))

Итоги

Isolates - Lifecycle

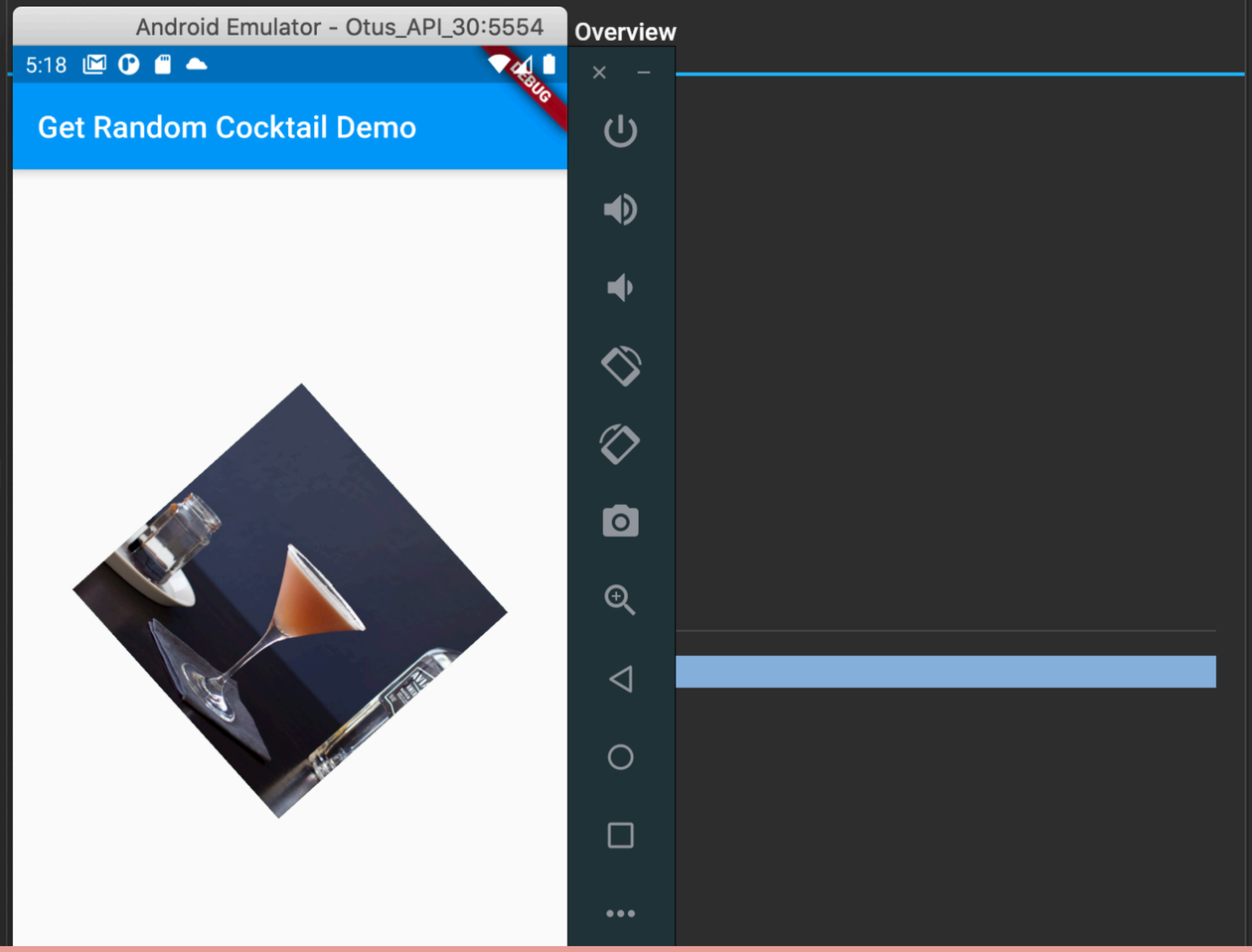
Isolates - Обмен данными

AsyncCocktailRepository in Isolate - Demo

Dart DevTools | Flutter Inspector | Timeline | Memory | Performance | Debugger | Network | Logging | App Size

Record network traffic | Stop | Clear | Search

Method	Uri	Status	Type	Duration	Timestamp
GET	InternetAddress('35.158.177.251', IPv4)	101	ws	6120 ms	5:18:06.181 PM
GET	https://the-cocktail-db.p.rapidapi.com/...	--	http	Pending	5:18:14.728 PM
GET	https://www.thecocktaildb.com/image...	--	http	18528 ms	5:18:15.834 PM
GET	InternetAddress('35.158.177.251', IPv4)	101	ws	Pending	5:18:16.078 PM
GET	InternetAddress('172.67.206.201', IPv4)	101	ws	Pending	5:18:19.564 PM
GET	https://the-cocktail-db.p.rapidapi.com/...	--	http	16900 ms	5:18:24.754 PM
GET	InternetAddress('35.158.177.251', IPv4)	101	ws	15473 ms	5:18:26.203 PM
GET	https://the-cocktail-db.p.rapidapi.com/...	--	http	3499 ms	5:18:34.726 PM
GET	InternetAddress('18.196.60.97', IPv4)	101	ws	2424 ms	5:18:35.812 PM
GET	https://www.thecocktaildb.com/image...	200	image/...	4070 ms	5:18:38.250 PM
GET	https://www.thecocktaildb.com/image...	200	image/...	4411 ms	5:18:41.698 PM
GET	InternetAddress('104.24.107.74', IPv4)	101	ws	Pending	5:18:42.835 PM
GET	https://the-cocktail-db.p.rapidapi.com/...	--	http	4617 ms	5:18:44.737 PM
GET	InternetAddress('35.158.177.251', IPv4)	101	ws	3341 ms	5:18:46.016 PM
GET	https://www.thecocktaildb.com/image...	200	image/...	6971 ms	5:18:49.385 PM
GET	https://the-cocktail-db.p.rapidapi.com/...	--	http	Pending	5:18:54.732 PM



Profiling Flutter Applications by Observatory

Observatory > vm@ws://127.0.0.1:64505/txIU_5WIhIk=/ws > main > allocation profile

Auto-refresh on GC | Refresh | GC | Download

Allocation Profile

last forced GC at ---

New Generation

used	3.5MB of 8.0MB
external	2.6MB
collections	53
average collection time	24.06 ms

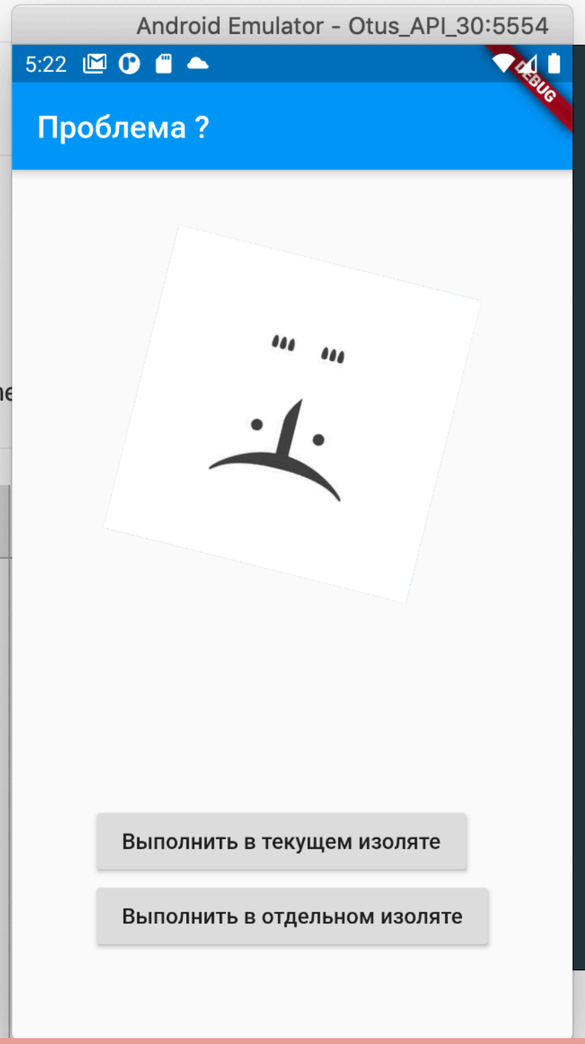
Old Generation

used	48.6MB of 51.2MB
external	301.7KB
collections	2
average collection time	compact ▲

Total

used	
external	
collections	
average collection time	

New Generation				Old Generation				Total			
Internal	External	Size	Instances	Internal	External	Size	Instances	Internal	External	Size▲	Instances
77.1KB	0B	77.1KB	1823	16.9MB	0B	16.9MB	76812	17.0MB	0B	17.0MB	78635
170.5KB	0B	170.5KB	3917	6.9MB	0B	6.9MB	144674	7.1MB	0B	7.1MB	148591
0B	0B	0B	0	6.1MB	0B	6.1MB	70	6.1MB	0B	6.1MB	70
0B	0B	0B	0	4.0MB	0B	4.0MB	8682	4.0MB	0B	4.0MB	8682
0B	0B	0B	0	2.7MB	0B	2.7MB	29669	2.7MB	0B	2.7MB	29669
6.3KB	2.3MB	2.3MB	810	144B	52.7KB	52.9KB	18	6.5KB	2.4MB	2.4MB	828
0B	0B	0B	0	1.8MB	0B	1.8MB	47864	1.8MB	0B	1.8MB	47864
1.4MB	0B	1.4MB	45394	17.3KB	0B	17.3KB	553	1.4MB	0B	1.4MB	45947
0B	0B	0B	0	1.3MB	0B	1.3MB	8682	1.3MB	0B	1.3MB	8682
0B	0B	0B	0	1.3MB	0B	1.3MB	41297	1.3MB	0B	1.3MB	41297
0B	0B	0B	0	1.2MB	0B	1.2MB	8025	1.2MB	0B	1.2MB	8025
40.6KB	0B	40.6KB	449	1.1MB	0B	1.1MB	84	1.2MB	0B	1.2MB	533
0B	0B	0B	0	993.2KB	0B	993.2KB	8114	993.2KB	0B	993.2KB	8114



Profiling Flutter Applications by Observatory

Observatory > vm@ws://127.0.0.1:8181/oJOBYoS92Ys=/ws > BugData Isolate > allocation profile

Auto-refresh on GC

Allocation Profile

last forced GC at ---

New Generation

used	1.0MB of 8.0MB
external	0B
collections	2
average collection time	0.33 ms

Old Generation

used	25.3MB of 26.8MB
external	32B
collections	3
average collection time	compact ▲

Total

used	26.3MB of 34.8MB
external	32B
collections	5
average collection time	0.33 ms

New Generation				Old Generation				Total				
Internal	External	Size	Instances	Internal	External	Size	Instances	Internal	External	Size▲	Instances	Class
7.4KB	0B	7.4KB	200	21.4MB	0B	21.4MB	8676	21.4MB	0B	21.4MB	8876	_OneByteString
15.3KB	0B	15.3KB	180	1.3MB	0B	1.3MB	18962	1.3MB	0B	1.3MB	19142	_List
0B	0B	0B	0	1.0MB	0B	1.0MB	6563	1.0MB	0B	1.0MB	6563	Function
0B	0B	0B	0	286.0KB	0B	286.0KB	2615	286.0KB	0B	286.0KB	2615	Field
0B	0B	0B	0	227.4KB	0B	227.4KB	485	227.4KB	0B	227.4KB	485	Instructions
0B	0B	0B	0	220.4KB	0B	220.4KB	1085	220.4KB	0B	220.4KB	1085	Class
0B	0B	0B	0	206.3KB	0B	206.3KB	5	206.3KB	0B	206.3KB	5	_Uint32List
0B	0B	0B	0	115.3KB	0B	115.3KB	1845	115.3KB	0B	115.3KB	1845	_Type
0B	0B	0B	0	105.0KB	0B	105.0KB	1680	105.0KB	0B	105.0KB	1680	ICData
1.2KB	0B	1.2KB	17	99.3KB	0B	99.3KB	232	100.5KB	0B	100.5KB	249	_Int8List

Прочее

Profiling Flutter Applications by Observatory

Observatory > vm@ws://127.0.0.1:8181/oJOBYoS92Ys=/ws > BugData Isolate > _OneByteString

Instances

currently allocated count 9246 (shallow size 222.6MB)

strongly reachable { ⊟
'org-dartlang-sdk:///sdk/lib/_http/http_date.dart'
'dart:_http/http_date.dart'
'org-dartlang-sdk:///sdk/lib/_http/http_session.dart'
'dart:_http/http_session.dart'
'org-dartlang-sdk:///sdk/lib/_http/http_headers.dart'
'dart:_http/http_headers.dart'
'org-dartlang-sdk:///sdk/lib/_http/http_parser.dart'
'dart:_http/http_parser.dart'
'org-dartlang-sdk:///sdk/lib/_http/websocket.dart'
'dart:_http/websocket.dart'
'org-dartlang-sdk:///sdk/lib/_http/crypto.dart'
'dart:_http/crypto.dart'
'org-dartlang-sdk:///sdk/lib/_http/websocket_impl.dart'
'dart:_http/websocket_impl.dart'
'org-dartlang-sdk:///sdk/lib/_http/http_impl.dart'
'dart:_http/http_impl.dart'
'org-dartlang-sdk:///sdk/lib/_http/http.dart'
'org-dartlang-sdk:///sdk/lib/_http/overrides.dart'
'dart:_http/overrides.dart'
'HttpStatus'
'valueOfNonNullableParamWithDefault'
'Since'
'log'
'UnmodifiableMapView'
'LinkedListEntry'
'LinkedList'
'ListQueue'
'Queue'

Profiling Flutter Applications by Observatory

Настройки Dart VM и GC

old_gen_growth_rate: 280 (The max number of pages the old generation can grow at a time)
old_gen_growth_space_ratio: 20 (The desired maximum percentage of free space after old gen GC)
old_gen_growth_time_ratio: 3 (The desired maximum percentage of time spent in old gen GC)
old_gen_heap_size: 30720 (Max size of old gen heap size in MB, or 0 for unlimited,e.g: --old_gen_heap_size=1024 allows up to 1024MB old gen heap)

Flutter: Don’t Fear the Garbage Collector
Matt Sullivan
https://medium.com/flutter/flutter-dont-fear-the-garbage-collector-d69b3ff1ca30

Заполните, пожалуйста,
опрос о занятии по ссылке в чате

Спасибо за внимание!
Приходите на следующие вебинары

Смирнов Андрей wrike

Курс **Мобильная разработка на Flutter**